FOUNDATION LESSON

WEEK 2 · LESSON 7

Slicing and mutation

Slicing sequences / List methods and mutation

SLICING AND MUTATION

MENTAL MODEL

Slicing and mutation

A slice excludes its stop position. Append adds one object, extend adds multiple elements, and sort mutates a list.
sorted returns a new list; list.sort returns None. Appending a list is different from extending with its contents.

SLICES AND STEPS

CODE WALKTHROUGH W02-L07-S03

```
01  values = [10, 20, 30, 40, 50]
02  print(values[1:4])
03  print(values[::2])
```

WHAT TO TRACK

Slices and steps

PREDICT THE RESULT

Predict the output.
Explain each operation.
Pause before continuing.

SLICES AND STEPS

CODE WALKTHROUGH W02-L07-S04

```
01 values = [10, 20, 30, 40, 50]
02 print(values[1:4])
03 print(values[::2])
```

WHAT TO TRACK

Slices and steps

VALIDATE THE RESULT

```
[20, 30, 40]
[10, 30, 50]
```

ADD AND SORT

CODE WALKTHROUGH W02-L07-S05

```
01  transactions = [100, 250]
02  transactions.append(500)
03  transactions.extend([75, 125])
04  transactions.sort()
05  print(transactions)
```

WHAT TO TRACK

Add and sort

PREDICT THE RESULT

Predict the output.
Explain each operation.
Pause before continuing.

ADD AND SORT

CODE WALKTHROUGH W02-L07-S06

```
01  transactions = [100, 250]
02  transactions.append(500)
03  transactions.extend([75, 125])
04  transactions.sort()
05  print(transactions)
```

WHAT TO TRACK

Add and sort

VALIDATE THE RESULT

```
[75, 100, 125, 250, 500]
```

APPEND OR EXTEND

CODE WALKTHROUGH W02-L07-S07

```
01  first = [10]
02  first.append([20, 30])
03  second = [10]
04  second.extend([20, 30])
05  print(first)
06  print(second)
```

WHAT TO TRACK

Append or extend

PREDICT THE RESULT

Predict the output.
Explain each operation.
Pause before continuing.

APPEND OR EXTEND

CODE WALKTHROUGH W02-L07-S08

```
01  first = [10]
02  first.append([20, 30])
03  second = [10]
04  second.extend([20, 30])
05  print(first)
06  print(second)
```

WHAT TO TRACK

Append or extend

VALIDATE THE RESULT

```
[10, [20, 30]]
[10, 20, 30]
```

SORT OR SORTED

CODE WALKTHROUGH W02-L07-S09

```
01  amounts = [30, 10]
02  ordered = sorted(amounts)
03  print(amounts)
04  result = amounts.sort()
05  print(ordered)
06  print(amounts)
07  print(result)
```

WHAT TO TRACK

Sort or sorted

PREDICT THE RESULT

Predict the output.
Explain each operation.
Pause before continuing.

SORT OR SORTED

CODE WALKTHROUGH W02-L07-S10

```
01  amounts = [30, 10]
02  ordered = sorted(amounts)
03  print(amounts)
04  result = amounts.sort()
05  print(ordered)
06  print(amounts)
07  print(result)
```

WHAT TO TRACK

Sort or sorted

VALIDATE THE RESULT

```
[30, 10]
[10, 30]
[10, 30]
None
```

POP AND MUTATION

CODE WALKTHROUGH W02-L07-S11

```
01  values = [10, 20]
02  removed = values.pop()
03  print(removed)
04  print(values)
```

WHAT TO TRACK

Pop and mutation

PREDICT THE RESULT

Predict the output.
Explain each operation.
Pause before continuing.

POP AND MUTATION

CODE WALKTHROUGH W02-L07-S12

```
01  values = [10, 20]
02  removed = values.pop()
03  print(removed)
04  print(values)
```

WHAT TO TRACK

Pop and mutation

VALIDATE THE RESULT

```
20
[10]
```

INDEPENDENT PREDICTION

LEARNER PRACTICE

Independent prediction

Change one input in slicing and mutation and test a boundary or edge case.

01 Predict the exact output.
02 Change only one input.
03 Run in a fresh Python process.
04 Explain any difference.

PREDICT · MODIFY · RUN

01 Restate the supplied rules.

02 Predict the result.

03 Modify and run the code.

04 Compare actual output.

05 Explain any difference.

SLICING AND MUTATION

Recap and next step

A slice excludes its stop position. Append adds one object, extend adds multiple elements, and sort mutates a list. sorted returns a new list; list.sort returns None. Appending a list is different from extending with its contents.

UNDERSTAND APPLY VALIDATE EXPLAIN